

Модель знание-емкого бизнес-процесса для задач интеллектуального анализа процессов

Предложена модель знание-емкого бизнес-процесса, включающая в себя набор его состояний и правила перехода между последними. Модель отличается от существующих тем, что связывает контекст с последовательностью действий процесса посредством темпоральных правил, что позволяет адаптировать модель во время выполнения бизнес-процесса путем включения новых правил.

Запропоновано модель знання-емного бізнес-процесу, яка містить набір його станів і правила переходів між ними. Модель відрізняється від існуючих тим, що пов'язує контекст і послідовність дій процесу за допомогою темпоральних правил, що дозволяє адаптувати модель під час виконання бізнес-процесу шляхом включення нових правил.

Введение. Интеллектуальный анализ процессов (*process mining*) направлен на построение и усовершенствование моделей функционирования взаимодействующих физических или информационных объектов на основе обработки информации о последовательностях событий, фиксирующих такое поведение [1, 2].

Результирующее поведение объектов представляется в виде модели, отражающей возможные (реализованные на практике) последовательности действий процесса использования и обработки указанных объектов. В качестве исходных данных, содержащих последовательности событий, чаще всего используются файлы логов (журналов регистрации событий) информационных систем, отражающих выполняющиеся на предприятии бизнес-процессы. Поэтому в практическом плане основное внимание в области *process mining* уделяется построению моделей, выполняющихся бизнес-процессов «как есть».

Знание-емкие (*knowledge intensive*) бизнес-процессы (ЗБП) [3] отличаются от традиционных процессов с предопределенным алгоритмом действий наличием уровня знаний, используемых для адаптации действий процесса во время выполнения с учетом текущего состояния предметной области [4]. Эти знания подразделяются по форме представления на явные и неявные [5]. Первые обычно представлены в формальном виде, а вторые – персональными знаниями исполнителей, отражающими их опыт и навыки [6]. Использование

исполнителями своих знаний для изменения хода ЗБП в зависимости от текущего контекста вызывает формирование новых последовательностей действий, которые не были включены в его исходную модель.

При моделировании бизнес-процессов традиционно основное внимание уделяется описанию последовательности действий процесса. Для формализации последовательности действий используются различные варианты *workflow*-моделей [7]. В таких моделях задается алгоритм действий процесса с указанием значений входных переменных, отражающих состояние предметной области. Явные знания о правилах выбора одного из вариантов реализации бизнес-процесса включаются в модель на этапе описания последовательности работ и не могут быть изменены во время выполнения процесса.

Альтернативный подход к моделированию бизнес-процессов заключается в построении моделей жизненного цикла объектов, с которыми взаимодействует бизнес-процесс [8]. Модель жизненного цикла фактически представляет собой *workflow*-модель фрагмента процесса, связанного с обработкой отдельного объекта. Однако в обоих рассмотренных подходах не учитывается наличие уровня знаний, существенно влияющего на ход выполнения процесса с учетом его текущего контекста.

Это приводит к возникновению проблемы спагетти – подобных графов при построении

workflow – модели ЗБП средствами process mining [1, 9]. Данная проблема состоит в том, что формируемый в результате интеллектуального анализа *workflow*-граф включает все возможные варианты реализации процесса без учета их значимости. В результате граф визуально напоминает блюдо спагетти: содержит избыточное количество вершин и дуг, затрудняющее понимание и использование модели бизнес-процесса аналитиком [10].

Поэтому решение задач построения и усовершенствования *workflow*-модели в результате выполнения интеллектуального анализа процессов, а также соответствующего повышения эффективности процессного управления сталкивается со значительными трудностями для ЗБП.

В связи с изложенным, проблема построения модели ЗБП, учитывающей зависимость действий процесса от контекста их выполнения, – актуальна.

Анализ исследований и публикаций

Исследования в сфере использования персональных знаний при управлении бизнес-процессами выполняются уже около двух десятилетий. Первоначально такие процессы рассматривались с позиций менеджмента и основное внимание уделялось организации работы работников, обладающих знаниями (*knowledge workers*), а также получению максимального эффекта от использования этих знаний. В соответствии с указанным подходом знание рассматривалось в первую очередь как результат работы такого бизнес-процесса, а не как средство обеспечения его нормального функционирования [11, 12].

Термин «знание-емкие (*knowledge-intensive*) бизнес-процессы» получил распространение в последнее десятилетие, после разработки общего подхода к описанию знание-емких бизнес-процессов с учетом неявной составляющей знаний, а также постоянной трансформации знаний между явной и неявной формами [3, 13].

В настоящее время существует ряд отличающихся подходов к определению ЗБП. При его описании выделяют следующие их ключевые особенности: использование явных и неявных знаний для получения результатов [4,

14]; сложность алгоритма, задающего последовательность действий в текущих условиях предметной области [15]; сотрудничество исполнителей при достижении цели процесса с использованием их персональных знаний [16]. Анализ указанных подходов показывает, что все основные свойства ЗБП, отличающие их от других классов бизнес-процессов, связаны с применением персональных знаний коллектива исполнителей для того, чтобы достичь цели процесса с максимальной эффективностью.

В *workflow*-модели ЗБП персональные знания отображаются парой (объект знаний, исполнитель). Объект знаний содержит краткую характеристику знаний, используемых исполнителем [3]. Знания в таком объекте характеризуются количественными показателями, отражающими степень использования опыта, профессиональных навыков, контекстных зависимостей [4]. Однако сами зависимости, влияющие на ход выполнения процесса, представлены в неявной форме и потому они не могут быть включены в формальную модель ЗБП. Этот недостаток не позволяет выделить все возможные варианты изменения ЗБП и использовать их для повышения эффективности процессного управления.

Для того чтобы включить неявные знания в модель, необходимо выполнить их экстернализацию (преобразование в явную форму). Формальные подходы к экстернализации неявных персональных знаний исполнителей пока разработаны недостаточно [6]. Поскольку неявные знания отражает применение личных персональных знаний людей при производстве продуктов, выполнении процессов, то оно может быть формализовано на основе анализа этих продуктов и процессов. Однако применяемые на практике подходы к его экстернализации направлены скорее на выявление некоторых идей, предположений и не рассматривают построение формализованных зависимостей [17, 18].

В то же время логи, используемые при решении задач интеллектуального анализа процессов, отражают действия исполнителей в конкретном контексте [1, 2, 19] и потому содержат встроенное знание. Это обуславливает принципиальную возможность экстернализации неявных знаний

ЗБП на основе анализа их логов, при условии использования моделей бизнес-процессов, включающих в себя компонент знаний.

Однако в настоящее время недостаточно исследованы вопросы, связанные с разработкой процессных моделей, включающих формализованные знания, и позволяющих дополнять набор этих знаний во время выполнения процесса с учетом текущего состояния процесса и его контекста.

Постановка задачи

Цель статьи состоит в построении модели ЗБП, которая включала бы в себя описание контекста выполнения, типовые последовательности действий, а также формализованные зависимости, обеспечивающие выбор действий процесса в зависимости от текущего состояния контекста.

Это даст возможность повысить эффективность процессного управления путем включения в контур управления экстернализованных знаний, отражающих опыт исполнителей.

Структура знание-емкого бизнес-процесса

Знание-емкий бизнес-процесс реализуется в форме итеративного выбора отдельных действий либо их последовательностей в текущем контексте. Выбор действий в общем случае осуществляется исполнителем на основе информации о текущем состоянии контекста, знаний о типовой последовательности работ и о влиянии контекста на ход процесса. Результирующий экземпляр процесса формируется в виде *workflow*-последовательности, полученной в результате взаимодействия исполнителей. Модель процесса включает все реализуемые в данном контексте экземпляры ЗБП.

Исходя из этого, предлагаемая авторами структура ЗБП включает в себя следующие составляющие:

- контекст выполнения, состоящий из набора объектов, используемых в ходе реализации процесса либо влияющих на ход выполнения процесса;
- знания о выполнении процесса, отражающие зависимости между контекстом и действиями процесса;

- последовательность действий процесса, реализуемая в условиях конкретного состояния контекста; совокупность возможных последовательностей для различных состояний контекста определяет *workflow*-описание знание-емкого процесса.

Приведенная структуризация показывает, что в разрабатываемой модели должны быть заданы элементы всех трех уровней, а также взаимодействие между этими элементами (рис. 1).



Рис. 1. Многоуровневая структура ЗБП

В то же время модель ЗБП должна обеспечивать возможность описания и верификации его поведения, а также возможность формализации выбора и выполнения последовательности действий на основе известных записей о поведении ЗБП.

Последняя возможность реализуется с использованием методов интеллектуального анализа процессов. Исходные данные для построения модели методами *process mining* представляются в виде набора последовательностей событий (*event*), называемых трассами (*trace*). Каждая трасса фиксирует выполнение одного

экземпляра процесса. Пример фрагмента трассы, представленный в стандарте описания логов XES, приведен на рис. 2.

```

<trace>
  <string key="concept:name" value="1-740866821"/>
  <event>
    <string key="org:group" value="N36"/>
    <string key="resource country" value="USA"/>
    <string key="organization country" value="us"/>
    <string key="org:resource" value="Matt"/>
    <string key="organization involved" value="Org line C"/>
    <string key="org:role" value="V3_2"/>
    <string key="concept:name" value="Accepted"/>
    <string key="impact" value="High"/>
    <string key="product" value="PROD321"/>
    <string key="lifecycle:transition" value="In Progress"/>
    <date key="time:timestamp" value="2012-05-05T14:54:27+02:00"/>
  </event>
  <event>
    ...
  </event>
  ...
</trace>

```

Рис. 2. Исходные данные для интеллектуального анализа процессов

Приведенный фрагмент исходных данных позволяет структурировать описание ЗБП в логе. Каждое выполнение экземпляра процесса отражается в логе в виде трассы. Действие процесса фиксируется в логе в виде одного или последовательности из нескольких событий. Последовательность событий трассы отражает выполнение последовательности действий для одного экземпляра процесса. Каждое событие, помимо временной метки *time:timestamp*, характеризуется набором переменных, содержащих значения атрибутов объектов, с которыми взаимодействует процесс. Так, в примере на рис. 2 переменная *org:resource*, отражающая используемый ресурс организации, содержит имя исполнителя *Matt*, переменная *org:role* – код роли исполнителя в текущем процессе, а переменная *product* – код обрабатываемого продукта. Действия процесса также описаны в виде значений переменных. В частности, переменная *concept:name* отражает

выполненное действие *Accepted*. Обобщая, описание событий на трассе процесса характеризует состояние используемых процессом объектов. Иными словами, информация о структуре и состоянии контекста процесса, а также об изменении состояния контекста содержится в последовательности событий лога.

Исходный лог характеризуется следующими особенностями:

- текущее состояние бизнес-процесса задается через набор значений атрибутов объектов контекста, связанных с каждым событием лога, и содержит условия для запуска действий процесса;
- последовательность переходов между состояниями, записанная в логе в виде последовательности событий, отражает выполнение действий процесса;
- связи между контекстом и действиями, влияющие на выбор действий в ходе процесса, скрыты в структуре лога в виде связей между атрибутами событий, характеризующих используемые объекты, а также атрибутами действий процесса.

Модель знание-емкого бизнес-процесса

Исходя из изложенных предпосылок, модель знание-емкого процесса представим как модель Крипке [20] \mathcal{M}_{BP} над множеством условий запуска действий процесса C :

$$\mathcal{M}_{BP} = (S, D, F) \quad (1)$$

где S – конечное множество состояний ЗБП; D – отношение переходов между состояниями; $F: S \rightarrow 2^C$ – отображение, которое с каждым состоянием $s \in S$ связывает подмножество условий $c \in C$, истинных в этом состоянии.

Отношение переходов между состояниями реализуется посредством выполнения действий ЗБП:

$$D \subseteq S \times S : \forall (s', s'') \in S \exists d(s', s'') \in D, \quad (2)$$

где s', s'' – произвольная пара состояний контекста бизнес-процесса; $d(s', s'')$ – действие процесса, которое переводит контекст из состояния s' в состояние s'' .

Для реализации одного из действий процесса $d \in D$ в произвольном состоянии $s \in S$ необходимо, чтобы выполнялось условие запуска c , входящее в состав правила вида (условие,

действие), т.е. $r = (c, d)$. Данное правило задает порядок выполнения лишь одного действия знание-емкого процесса.

В то же время анализ особенностей ЗБП показал, что они – обобщение традиционных бизнес-процессов с априорно заданной структурой. ЗБП объединяет фрагменты процесса, требующие принятия решений исполнителем и фрагменты с предопределенной последовательностью действий. В этом случае цикл выполнения ЗБП включает в себя шаги: выбор последовательности действий; поочередное выполнение действий до завершения выбранной последовательности; выбор новой последовательности и т.д. Для начального действия последовательности существует правило $r_0 = (c_0, d_0)$, а для всех последующих – $r_i = (true, d_i)$.

По мере осуществления выбранной последовательности действий выполняется безусловный переход между состояниями, при котором изменяются как значения атрибутов объектов контекста, так и взаимосвязи между объектами. Изменяемые значения переменных и связей образуют постусловие выполненной последовательности действий. Поэтому при описании поведения ЗБП необходимо перейти от правил в форме (условие, действие) к правилам в форме (предусловие, постусловие). В дальнейшем такие правила будем называть подпроцессами, поскольку они содержат как определенную область знаний, так и *workflow*-составляющую.

Определение. Под подпроцессом g в модели поведения ЗБП M_{ZP} будем понимать пару (предусловие c , постусловие d), где предусловие представляет собой логическую формулу состояния, отражающую состояние контекста до выполнения подпроцесса, а постусловие представляет собой логическую формулу пути, отражающую последовательность действий и результаты выполнения подпроцесса.

Для представления формул состояния и пути используем темпоральные логики [20], поскольку последние представляют собой формализм, применяемый для описания последовательности переходов реагирующей системы без явного упоминания времени. В состав таких логик входят темпоральные операторы, позволяющие ука-

зать возможность достижения интересующего авторов состояния. При моделировании бизнес-процессов темпоральная логика позволяет показать достижимость (недостижимость) состояния, свидетельствующего о достижении цели ЗБП, что обеспечивает возможность последующей верификации модели процесса.

В данной статье используется темпоральная модальная логика CTL^* – логика ветвящегося времени. Данная логика позволяет описать дерево реализации бизнес-процесса, содержащее все возможные экземпляры ЗБП, запускаемые из некоторого начального состояния. Использование этой логики обусловлено ключевой особенностью ЗБП, представленной в структурной модели: изменение хода выполнения процесса путем выбора исполнителей одного из возможных действий (последовательностей действий). В различных экземплярах ЗБП могут быть реализованы альтернативные последовательности действий во времени, что приводит к необходимости использования логики ветвящегося времени. Отметим, что альтернативная логика линейного времени LTL позволяет описать только один экземпляр бизнес-процесса, поскольку такой экземпляр содержит только одну выполнившуюся последовательность действий.

Для описания поведения ЗБП как последовательности переходов между состояниями определим семантику формул предусловий и постусловий подпроцесса, а также связь между ними в модели Крипке.

Формулы предусловий c_i вычисляются на основе информации о текущем состоянии контекста в виде текущих значений атрибутов объектов, связей между атрибутами и зависимостей между значениями атрибутов и потому есть формулами состояния. Их истинность в модели M_{ZP} обозначается для некоторого состояния s символом \models .

Истинность (либо неистинность) произвольного предусловия c_i , а также истинность конъюнкции и дизъюнкции предусловий c_i и c_j в модели M_{ZP} в состоянии s определяется традиционно:

$$M_{ZP}, s \models c_i \Leftrightarrow c_i \in F(s);$$

$$M_{ZP}, s \models \neg c_i \Leftrightarrow M_{ZP}, s \not\models c_i;$$

$$\begin{aligned} M_{ZP}, s \models c_i \wedge c_j &\Leftrightarrow M_{ZP}, s \models c_i \wedge M_{ZP}, s \models c_j; \\ M_{ZP}, s \models c_i \vee c_j &\Leftrightarrow M_{ZP}, s \models c_i \vee M_{ZP}, s \models c_j. \end{aligned} \quad (3)$$

Связь между некоторым состоянием s и путем выполнения π для произвольного экземпляра процесса задается следующим образом:

$$M_{ZP}, s \models Ec_i \Leftrightarrow \exists \pi^s : M_{ZP}, \pi^s \models c_i, \quad (4)$$

где π^s – часть полного пути выполнения процесса π , которая начинается с состояния s ; E – квантор пути, задающий возможность реализации действий в экземпляре процесса.

Выражение (4) задает возможность достижения некоторого предусловия c_i на данном пути выполнения процесса π , начиная с текущего состояния s . Неформально это означает, что авторы статьи *на правильном пути* и, после выполнения некоторого предопределенного количества действий, получим истинное предусловие c_i для выполнения других действий процесса. Следовательно, выбранная последовательность действий, которая привела процесс в состояние s , должна быть продолжена до достижения состояния, в котором выполняется условие c_i .

Отметим, что полный путь выполнения процесса π представлен в логике в виде трассы. Поэтому выражение (4) при решении задач интеллектуального анализа процессов определяет наличие на трассе процесса такого события, для которого c_i будет истинным.

Определим связь между некоторым состоянием s и множеством путей выполнения π для всех экземпляров процесса:

$$M_{ZP}, s \models Ac_i \Leftrightarrow \forall \pi^s \in \Pi M_{ZP}, \pi^s \models c_i, \quad (5)$$

где Π – множество всех путей реализации ЗБП, включая как полные пути π , так и неполные пути π^s , т.е. $\pi, \pi^s \in \Pi$; A – квантор пути, задающий возможность реализации действий во всех экземплярах процесса.

Семантика выражения (5) состоит в том, что из данного состояния s можно выбирать любой путь для того, чтобы обеспечить истинность формулы c_i . Иными словами, в данном состоянии на достижение результата c_i не влияют знания исполнителя, выбирающего дальнейшую последовательность действий бизнес-процесса.

Определим постуловия d_i в модели M_{ZP} . Постуловия задают реализацию подмножеств действий некоторого подпроцесса g_i , а также результаты этих действий в виде изменения контекста на конкретном пути реализации процесса. Поэтому формулы постуловий есть формулами пути.

Истинность произвольного постуловия d_i на части пути π выполнения процесса определяется традиционно:

$$M_{ZP}, \pi^s \models d_i \Leftrightarrow M_{ZP}, s \models d_i. \quad (6)$$

В этом случае истинность постуловия d_i определяется для пути реализации процесса, начиная с состояния s . Это означает, что подпроцесс g_i с постуловием d_i полностью завершен в состоянии s .

Связь нескольких последовательно выполняющихся процессов определяется через постуловие следующим образом:

$$M_{ZP}, \pi \models d_i \Leftrightarrow M_{ZP}, s_0 \models d_i \mid \pi^{s_0} = \pi, \quad (7)$$

где s_0 – начальное состояние пути π .

В данном случае результаты предшествующего ЗБП используются при запуске последующего процесса, поскольку истинность постуловия d_i в начальном состоянии s_0 пути π означает, что данная формула была выполнена в результате реализации предшествующего процесса. Поэтому d_i можно рассматривать как предусловие для текущего процесса, начиная с начального состояния s_0 .

Неистинность произвольного постуловия d_i означает, что подпроцесс g_i не может быть выполнен на пути π :

$$M_{ZP}, \pi \models \neg d_i \Leftrightarrow M_{ZP}, \pi \not\models d_i \quad (8)$$

При решении задач интеллектуального анализа процессов это означает, что текущая трасса не содержит записи о выполнении действий d_i .

Для объединения результатов различных подпроцессов определим конъюнкцию и дизъюнкцию их постуловий d_i и d_j в модели M_{ZP} для пути π :

$$\begin{aligned} M_{ZP}, \pi \models d_i \wedge d_j &\Leftrightarrow M_{ZP}, \pi \models \\ &\models d_i \wedge M_{ZP}, s \models d_j \mid d_i \neq d_j, \\ M_{ZP}, \pi \models d_i \vee d_j &\Leftrightarrow M_{ZP}, \pi \models \end{aligned} \quad (9)$$

$$\models d_i \vee \mathcal{M}_{ZP, s} \models d_j \mid d_i \neq d_j, \quad (10)$$

где d_i, d_j – различные постусловия, истинные на одном и том же пути π , т.е. принадлежащие одному экземпляру бизнес-процесса.

Постусловия представляют знания о выполнении подпроцесса в случае их истинности. С каждым элементом постусловия связан переход между состояниями текущего экземпляра бизнес-процесса. Поэтому ограничение $d_i \neq d_j$ не позволяет дважды выполнить идентичные постусловия (т.е. идентичные последовательности действий) на одном и том же пути реализации процесса.

Реализуемость подпроцесса g_i в данном экземпляре бизнес-процесса задается посредством определения истинности его действий d_i в одном из последующих состояний пути π .

Выполнение последнего действия подпроцесса $g_i = (c_i, d_i)$ (или подпроцесса из одного действия) определяется следующим образом:

$$\mathcal{M}_{ZP, \pi^s} \models Xd_i \Leftrightarrow \mathcal{M}_{ZP, \pi^{s+1}} \models d_i, \quad (11)$$

где π^s, π^{s+1} – фрагменты пути π , начинающиеся с состояний s и $s+1$ соответственно; оператор X позволяет задать истинность произвольного постусловия d_i в состоянии, непосредственно следующим за текущим.

Реализуемость подпроцесса $g_i = (c_i, d_i)$ для данного экземпляра бизнес-процесса без указания состояния завершения определяется следующим образом:

$$\mathcal{M}_{ZP, \pi^s} \models Fd_i \Leftrightarrow \exists l > 0: \mathcal{M}_{ZP, \pi^{s+l}} \models d_i, \quad (12)$$

где l – количество действий процесса, которые должны быть реализованы, чтобы выполнилось постусловие d_i ; π^s, π^{s+1} – фрагменты пути π , начинающиеся с состояний s и $s+1$ соответственно; F – оператор, который задает истинность произвольного постусловия d_i на текущем пути в одном из последующих состояний.

Выполнимость подпроцесса для всех последующих состояний, начиная с текущего состояния, задается следующим образом:

$$\mathcal{M}_{ZP, \pi^s} \models Gd_i \Leftrightarrow \mathcal{M}_{ZP, \pi^{s+l}} \models d_i \forall l > 0, \quad (13)$$

где оператор логики G задает истинность d_i на текущем пути во всех последующих состояниях.

Реализуемость подпроцесса g_i только при условии выполнения подпроцесса g_i задается следующим образом:

$$\mathcal{M}_{ZP, \pi^s} \models d_i U d_j \Leftrightarrow (\exists l: \mathcal{M}_{ZP, \pi^{s+l}} \models d_j) \wedge (\mathcal{M}_{ZP, \pi^{s+m}} \models d_i \mid m < l), \quad (14)$$

где $s+m, s+l$ – индексы состояний на пути π ; оператор U определяет истинность произвольного постусловия d_i на текущем пути после выполнения предусловия во всех предшествующих состояниях.

Представление знаний знание-емкого бизнес-процесса

Представленная семантика логического описания ЗБП позволяет определить правила, обеспечивающие решение задач поддержки выполнения бизнес-процесса, а также оценки достижения целей процесса. Такие зависимости включают в себя: правила выполнения действий, синхронизации; объединения результатов, поддержки принятия решений и т.д. Совокупность таких правил определяет модель представления знаний ЗБП

Например, правило $EcXd$ задает доступность в текущем состоянии такого пути, на котором будет истинным условие c , после чего реализуется действие d , правило $AcXd$ показывает, что на всех путях из текущего состояния будет истинным условие c , после чего выполнится действие d .

При решении задач *process mining* первое правило задает существование такой трассы процесса, в составе которой существует пара событий с зависимостью между атрибутами cXd . Тогда первое событие содержит набор значений переменных, определяющих предусловие c , а второе событие содержит переменные, определяющие постусловие d .

Второе правило показывает, что при достижении текущего состояния на всех трассах процесса будет пара событий, задающих cXd .

Отметим, что рассмотренные правила могут включать в себя экстернализованные неявные зависимости, формируемые с использованием методов интеллектуального анализа процессов.

Заключение. Предложена типовая модель ЗБП, которая задает логическое описание множества его состояний, а также поведения процесса. Состояния бизнес-процесса характеризуются набором атрибутов объектов, используемых процессом. Модель отличается от существующих процессных моделей тем, что содержит компоненту знаний, определяющую отношения переходов между состояниями в форме набора темпоральных правил выбора действий процесса в зависимости от состояния контекста. Для представления знаний используются средства темпоральной логики, поскольку такие зависимости должны задавать адаптируемый во времени алгоритм действий процесса. Представление знаний для конкретной предметной области формируется путем выявления множества логических темпоральных правил средствами интеллектуального анализа процессов.

Последовательность применения правил задает *workflow*-описание ЗБП.

Формализация типового описания ЗБП на основе темпоральной логики обеспечивает возможности адаптации модели во время выполнения ЗБП путем ее дополнения новыми темпоральными правилами, а также верификации поведения процесса с использованием известных методов верификации моделей программ (*model checking*).

Практическая значимость разработанной модели связана с возможностью включения экстернализованных методами интеллектуального анализа неявных знаний в модель процесса и последующего использования этих знаний для повышения эффективности процессного управления.

1. *W.M.P. van der Aalst*. Process Mining: Data Science in Action. Aalst. – Berlin: Springer-Verlag, 2016. – 466 p.
2. *W.M.P. van der Aalst*. Process Mining in the Large: A Tutorial // Business Intelligence. – Springer Science + Business Media, 2014. – P. 33–76.
3. *Gronau N*. Modeling and Analyzing knowledge intensive business processes with KMDL: Comprehensive insights into theory and practice (Engl). – Gito, 2012. – 522 p.
4. *A Proposal to Model Knowledge in Knowledge-Intensive Business Processes* / N. Gronau, C. Thim, A. Ullrich et al. // Proc. of the Sixth Int. Symp. on Business Modeling and Software Design. – 20–22 June. – Rhodes, Greece. BMSD 2016. – **16**. – P. 98–103.
5. *Collins H*. Tacit and Explicit Knowledge. – The Univ. of Chicago Press. – 2010. – 186 p.

6. *Nonaka И., Takeuchi Х*. Компания – создатель знания. Зарождение и развитие инноваций в японских фирмах. – М.: Олимп-Бизнес, 2003. – 384 с.
7. *Weske M*. Business Process Management: Concepts, Languages, Architectures. – Berlin – Heidelberg: Springer-Verlag, 2012. – 403 p.
8. *Cohn D*. Business artifacts: A data-centric approach to modeling business operations and processes // Bulletin of the IEEE Computer Society Technical Committee on Data Engineering. – 2009. – **32**, № 3. – P. 1–7.
9. *Handling Concept Drift in Process Mining* / Jagadeesh Chandra Bose R.P., W.M.P. van der Aalst, I. Zliobaite et al. // Int. Conf. on Advanced Information Syst. Engin. (Caise 2011). Lecture Notes in Computer Science. – Berlin: Springer-Verlag, 2011. – **6741**. – P. 391–405.
10. *Visual Representations* / C. Görg, M. Poh, E. Qeli et al. // Human-Centered Visualization Environments. – Springer Science + Business Media, 2007. – **4417**. – P. 163–230.
11. *Davenport T., Jarvenpaa S., Beers M.*, Improving Knowledge Work Processes // Sloan Management Review, 1996. – P. 53–66.
12. *Davenport T*. Improving the Performance of People, Processes and Organizations. – Harvard Business School Press, 2005. – 227 p.
13. *Gronau N. Müller C., Korf R*. KMDL-Capturing, Analysing and Improving Knowledge-Intensive Business Processes // J. of Universal Computer Science. – 2005. – N 11 (4). – P. 452–472.
14. *Richter-von Hagen C., Ratz D., Povalej R*. Towards self-organizing knowledge intensive processes // J. of Universal Knowledge Management. – 2005. – **2**. – P. 148–169.
15. *Panian Z*. A promising approach to supporting knowledge intensive business processes: business case management // World Acad. of Sci., Engin. and Technol. – 2011. – **75**. – P. 642–648.
16. *Isik O., Mertens W., Van den Bergh J.*, Practices of knowledge intensive process management: quantitative insights // Business Process Management J. – 2013. – **19**, N 3. – P 515–534.
17. *Leonard-Barton D.A*. Wellsprings of Knowledge: Building and Sustaining the Sources of Innovation. – Boston: Harvard Business School Press, 1995. – 334 p.
18. *Nonaka I., Krogh G*. Tacit Knowledge and Knowledge Conversion: Controversy and Advancement in Organizational Knowledge Creation Theory // Organization Science. – May–June 2009. – **20**, N 3. – P. 635–652.
19. *W.M.P. van der Aalst*. Process Mining: Discovery, Conformance and Enhancement of Business Processes. – Berlin – Heidelberg: Springer, 2011. – 352 p.
20. *Кларк Э., Грамберг О., Пелед Д*. Верификация моделей программ: Model Checking. – М.: МЦНМО, 2002. – 416 с.

Поступила 11.11.2016

E-mail: levykinvictor@gmail.com, oksana.chala@nure.ua

© В.М. Левыкин, О.В. Чалая, 2016

Окончание на стр.72

UDC 004.891.3

V.M. Levykin, O.V. Chalaya

The Model of Knowledge-Intensive Business Process for the Process Mining

The concept drift problem of knowledge-intensive business process is considered. The problem lies in the fact that knowledge workers use their personal knowledge to change the workflow during its execution. They take into account the context of the process to change the workflow. The result is a complex structure of the workflow. That is why, for managing the business process, it is important to include knowledge that makes the concept drift in the process model. It gives the opportunity to make the process model more adequate to the real knowledge intensive business processes. So, the purpose of this paper is development of process model that includes the description of the process context, the workflow and the knowledge. That knowledge is used to choose an activity of a workflow in the specific context.

Model of knowledge-intensive business process for process mining is proposed. The model includes a set of states of a business process and a set of transition rules. State of the business process is defined by a set of attributes of objects that uses the business process. The relation of the transition between the states is determined by the logical rules. These rules define the possible workflows in accordance with the various conditions of the business process context.

The model differs from the existing ones by linking the context and workflow through the logical rules. This makes it possible to adapt the model of business process at runtime by integrating the new rules.

The model is designed to create a description of complex knowledge intensive processes using process mining techniques. Process mining techniques use the event logs to build a process model. State of the business process in the event logs is defined by a set of attributes of the events that belong to the log. We can use a subset of the event log for problems of process mining to build an interesting fragment of the proposed process model. This subset is defined by a set of pre-defined event attributes. These attributes belong to the events, and to a priori defined subset of objects that belongs to the context of the process. This allows to avoid building a spaghetti-like workflows.

